Politecnico di Torino

Digital systems electronics
A.A. 2018/2019

Prof. G. Masera

Lab 04

8 Apr 2019

Berchialla Luca	236032
Laurasi Gjergji	238259
Mattei Andrea	233755
Lombardo Domenico Maria	233959

1 Gated SR latch

The circuit shown in *figure* 1 implements a gated SR latch. Once coded into VHDL the Quartus Prime compiler uses separate memory components as shown in *figure* 2 to depict the electric circuit.

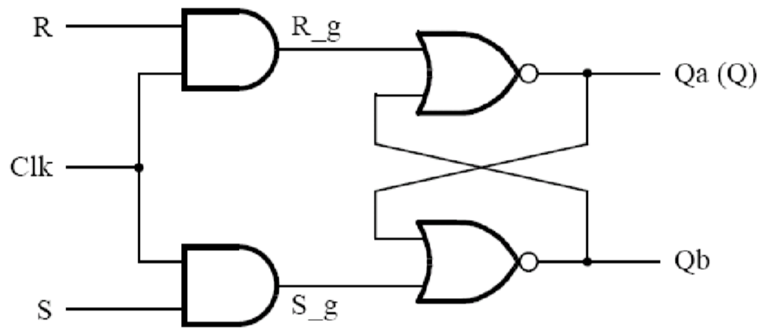

Figure 1: A gated SR latch circuit

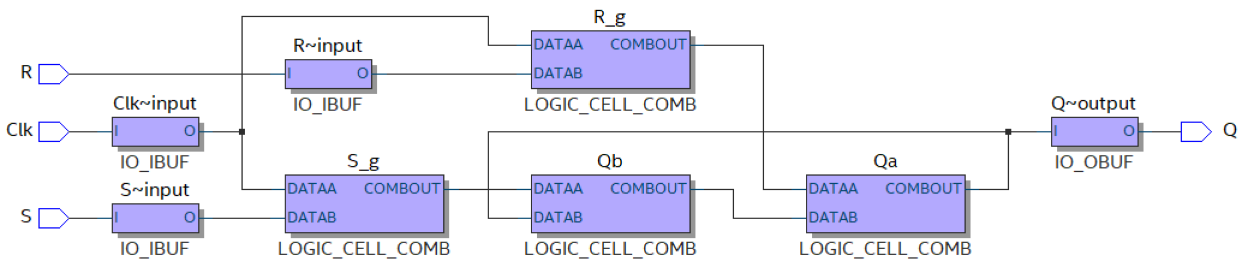

Figure 2: Quartus circuit implementation

Finally, a testbench has been written to test the behavior of the circuit, according to the table shown below:

Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$ (no change)
1	0	0	$Q(t)$ (no change)
1	0	1	0
1	1	0	1
1	1	1	x

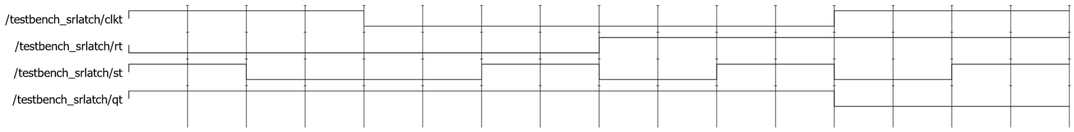

Figure 3: Modelsim Testbench waves

The testbench results are shown by means of the simulated waves shown in $figure$ 3. Initially $CLK = 1$, the lath is enabled. For $S = 1$, $R = 0$ the set condition is triggered and $Q = 1$. Viceversa $Q = 0$ when $S = 0$, $R = 1$, in reset condition. Once $CLK = 0$ the latch is disabled entering in the memory condition as $S = 0$, $R = 0$. In the other hand for $S = 1$, $R = 1$ the latch behavior becomes unpredictable.

2 16-bit synchronous counter

The circuit in $figure$ 4 implements a 4-bit counter using T flip flops. A 16-bit version has been implemented using the same structure as shown in figure 5. Using the Quartus tools the maximum working frequency has been identified to be equal to $F = 374.53MHz$ as reported in $figure$ 5 using a total of 31 LEs.

In the implementation of the 16-bits counter respect to the 4-bit counter no differences as observed in terms of architecture.

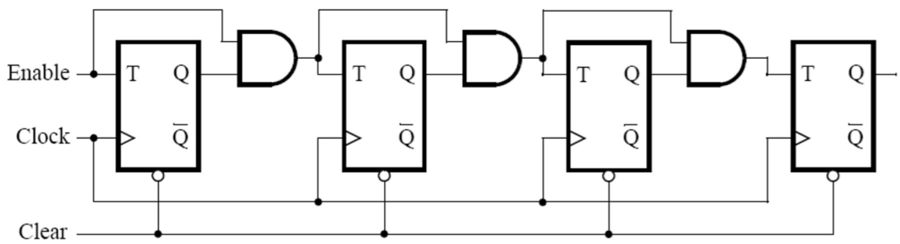

Figure 4: 4-bit counter

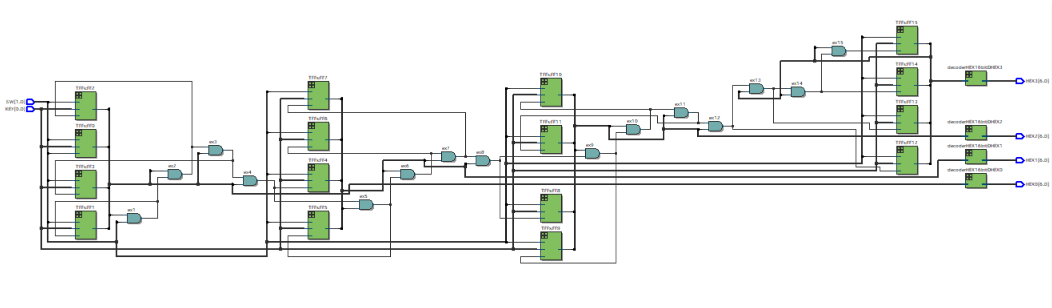

Figure 5: 4-bit counter

Fmax	Restricted Fmax	Clock Name
374.53 MHz	374.53 MHz	KEY0

Figure 6: Maximum working frequency

Finally a testbench has been designed to check the functionality of the circuit, the results are shown in $figure$ 6 where the counting process is shown using the HEX display. Notice that the circuit is firstly initialized resetting the current state of every FF. Then the counting process has been started enabling the circuit and applying a clock to every FF.

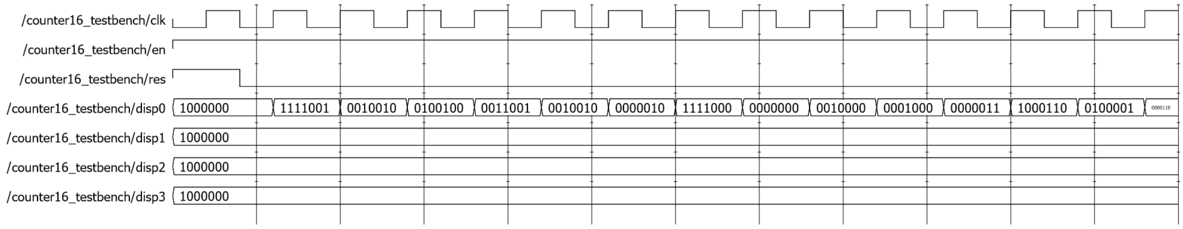

Figure 7: Modelsim Testbench waves

3 16-bit synchronous counter version 2

4 Flashing digits from 0 to 9

5 Reaction Timer

The RT level circuit to implement the reaction timer is shown in $figurex$. Some non standard components are just the implementation of more standard elements in a single component. This was done in order to accelerate the vhdl description of the circuit and simplify the block scheme.

5.1 millisecond counter

The purpose of this block is to generate a pulse on the output port connected to $millisecond_signal$ lasting for a clock cicle every millisecond. It has been implemented as a counter that resets and produces a pulse when it reaches the binary equivalent of 50000. With the DE1 $CLOCK\_50$ clock input at frequency $50MHz$ the circuit counts to 50000 in $1ms$.

5.2 counter comparator

This unit starts to count after it is resetted. Its enable signal is in *and* with the $millisecond_signal$ to enable the counter only every ms and not every clock cycle. When the counter reaches the value set by the inputs SW_{7-0} it stops and the $enable_out$ output is asserted to 1 until the component is reset.

5.3 BCD 4digits counter

This component is made by four $BCDcounters$ connected in a way that they count, modulus 10, numbers up to four digits. Each digit is represented binary using the BCD code and is connected to a decoder driving its respective 7-segments display. As shown in $figurexx$ each $BCDcounter$ has an input *enable* and output $enable_out$, those signals are connected together between consecutive $BCDcounters$. In particular the $enable_out$ output is asserted to 1 when the digit displayed is 9 in order to enable the following counter in the next period and properly display the number. The period for the counter is set to $1ms$ by taking the *and* of each enable input with the $millisecond_signal$.

5.4 Gated SR latches and connections between blocks

Two memory elements have been used to implement the requested behavior. The $MEM0$ element is responsible to generate the enable signal for the $counter_comparator$ when occurs pulse of KEY_0. The KEY_0 input is connected also to the $Asynchronous_clear$ input of the $counter_comparator$ in order to start to count from 0 when KEY_0 is pressed.
The $MEM1$ generate a signal useful to stop the $BCD_4digits_counter$ to count and turn off the $LEDR_0$. when KEY_3 is pressed. In particular the $LEDR_0$ is active from when the time set by the SW_{7-0} is eplaced (thanks $counter_comparator$) and up to when KEY_3 is pressed and the complemented output of $MEM1$ is 0. The $BCD_4digits_counter$ is active in the same amount of time, but since it has to count every millisecond its enable is in *and* with the $millisecond_signal$.

5.5 Testbench

To be able to verify easily the behaviour of the circuit a new version:"LAB4_ES5_tt has been created. This version does not include the 7-segments decoder, therefore the output is directly taken from the $BCD_4digits_counter$.
As first thing a clock signal with frequency $f = 50MHz$ was generated. Then two reaction time measurements cycles are executed with the following parameters:

Cycle	Waiting time	Reaction time
	3ms	17ms
	1ms	21ms

Relatively short duration have been chosen in order to make the simulation faster.